

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: TRAVIS C. RILEY

\$

FILING DATE: 12/20/01

\$

ATTORNEY DOCKET NO.  
3587/00-410

SERIAL NO. 09/745,262

\$

EXAMINER: CHARLES R. KYLE  
Art Unit 3624TITLE: METHOD AND APPARATUS FOR  
CREATION AND TRANSMISSION OF  
FINANCIAL STATEMENT DATA

\$ \$ \$ \$

DECLARATION OF JOHN C. HUGHES

Set forth below is the Declaration of John C. Hughes who, after being first cautioned and advised of the penalties of perjury, did declare as follows:

My Name is John C. Hughes. I reside at 5710 Dellwood Lane Beaumont, Texas 77706. I am a graduate of Lamar University, Beaumont Texas, having received a B.B.A., Bachelor Business Administration in Accounting, from that University in 1987. Since my graduation I have had extensive professional experience in the area of banking operations, computer software programs and computer programming for such operations. From 1987 through 1991 I was employed by Allied Bank, First Interstate Bank, Texas Commerce Bank, Chemical Bank and Chase Bank where my duties and responsibilities entailed a study of all banking operations and services and a computer operations and programs involving same. From 1991 through 2000, I was employed by First Bank and Trust and SouthTrust Bank. In 1991 - 1992 I was Loan Operations Manager. My duties and responsibilities included the establishment of a credit analytical analysis department; establishing a loan documentation production department; streamlining and increasing loan operations output; managing the merchant credit card services program; and managing the agent credit card program. From 1993 to 1995 I was the Loan Operations/Computer Network Manager. I was responsible for establishing a local area

computer network; establishing a wide area computer network, with remote locations; establishing computer host access, remote printing and remote computer assistance operations; and establishing computer automations solutions for deposit operations, loan operations, mortgage operations and other bank related programs. From 1995 through 1998 I was Loan Operations/Project Manager. During this time I built the overdraft protection product as well as implemented a 24-hour banking with voice or touch-tone computerized service. I also was involved in establishing a Visa check card program and assisted in restructuring and implementation of new deposit and loan products and pricing. From 1998 to 2000 I was a Project Manager for the bank where I was responsible for check imaging and direct on-line check processing services. Based upon my education and professional work experiences, I am familiar with typical and new products and services offered in the banking industry. I am also familiar with computer programs involved in the offering of such services and for the banking business, in general.

In the year 2000, I established ToGo Solutions, LLC, in Beaumont, Texas. ToGo Solutions, LLC is a limited liability company providing services to industries, including the banking industry and including the evaluation of software product functionality, feasibility and cost analysis involved in the preparation and updating of computerized programs for numerous activities involved in general banking operations. Representative clients of ToGo Solutions with whom I have performed such services include: Gulf Coast Bank; SouthTrust Bank; First Bank & Trust; Bridge City Bank; Security State Bank; Computer Specialist and EXP Internet Services.

I am familiar with United States Patent Application 09/745, 262, entitled "Method and Apparatus for Creation and Transmission of Financial Statement Data" filed December 20, 2001. I have read and studied the Office Action relating thereto, mailed July 15, 2004. I have studied the Blossman reference cited therein. I am familiar with and have used Microsoft Outlook 98.

Furthermore, I have reviewed and studied the claims as amended in this patent application. In my opinion, the claims as amended and as depended upon said amended claims are not obvious to one skilled in the art of computerized programs for the banking service industry, and as explained by the Examiner in the Office Action of July 15, 2004. The Blossman reference discloses method and apparatus for transmitting by electronic mail imagery as it would be sent to a customer, instead of via hard copy in standardized mail. In contrast, the method and apparatus of the present invention transmits data to the customer, which is pre-selected by the customer from all financial data available through the bank computer. This step and means are not obvious to me from a reading of the Blossman et al patent.

In my opinion, the currently amended claims are clearly distinguishable, not anticipated by, or obvious in view of the Blossman et al reference for at least the reason that the additional step of building an electronic file comprised of data extracted from the data pre-selected by the customer is not suggested or inherently disclosed therein. This is an important feature of the present invention as set forth in the original and currently amended claims in that reliance upon reports generated by the host financial institution through its computer system is eliminated, and the additional step of building an electronic file report as designated by the customer enables the transmission in real time increments designated by the customer of only the pre-selected data in the report file which is built. Nowhere in Blossman et al is this disclosed or inherently suggested.

In the present application, figures 1, 5, 13, 15, 17 and A each clearly exemplify a step and means for building an electronic file report comprised of data extracted from the data pre-selected by the customer and which can be generated at real time increments designated by the customer. See Fig. 1 for data which can be transmitted on any frequency desired by the customer not the host financial institution. Fig. 5 demonstrates an input screen reflecting the

transmitted pre-selected data from the report file on days other than when the financial institutions host computer system generates periodic statements. The step of building the electronic file report enables the customer the ability to set preferences for each sub-account. This is in contrast with Blossman et al which simply provides for the generation of a regular consolidated or unconsolidated periodic statement. The required "building" step and means are not disclosed or suggested or obvious in view of the Blossman et al reference.

Likewise, for example, fig. 15 in the application clearly shows the step for building a data base from the financial data in the computer system with respect to trial balance reports TBA1005 and 007. Extraction or building of the data base is further demonstrated in Figure 17 through the use of the posting journal PJL001.

I have attached certain pages from the specification of the present application. While the pages are unnumbered in the application as filed, in order to assist the Examiner I have numbered these source code pages, with the first page being the first page of the specification containing the source code. Pages 12 and 13 of the source code disclose the recordation of customer preferences as a load data sub-routine. Pages 15 and 16 again reference customer preferences as designated. This is the Cmdupdate feature. Pages 18, 19, 20, and 21 reference the VBtextBox fields for frequencies of statement transmission as designated by the customer. Page 23 of the source code references the VB combo. This demonstrates a selection of account types including a drop down box demonstrating that the intent of the programmer is to give a customer the ability to choose or select which accounts or portions of accounts for the transmission of a report and to designate or specify unique preferences by account. Pages 26, 27, 30, 31, and 32 reference check box codes and label codes and pages 40 and 41 disclose the broadcast message component, i.e., "transmitting the formatting pre-selected data", recited in the claims.

Likewise, on pages 44, 45 and 46 of the source code the extraction of loan information from a report file (Tba 1005. prn and Tba 1007. prn) are being built. As shown, these files are reports generated by core processor (FiServ). The reports would be used internally by the financial institution for balancing, research or to use in the host system where there is a breakdown. These pages clearly, in my opinion, distinguish a key feature of the applicant's invention in view of the Blossman et al reference which fails to disclose the extraction of information from report files to build a database so the forms can be built to conform to customer preferences.

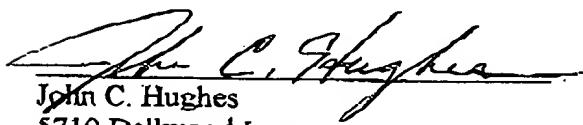
Additionally, page 64 of the source code (frmGenerate -9) shows the building of a database using loan information in the financial data in the financial institution computer system.

Pages 91 and 92 of the source code show a sub-routine for searching report files necessary to build the database tables and develop the pre-selected data to be transmitted customer preferences are clearly indicated versus standard periodic statements as shown in Blossman et al.

Finally, page 156 references the database named (ebalance.mdb) and page 157 of the source code, references tables in the database, named (ebalance.mdb). The extraction in building of this database or tables is important to the allowance of customized data pre-selected by the customer for transmission.

Further, Declarant Sayeth Not.

Signed under penalty of perjury this 14<sup>th</sup> day of January 2005.

  
John C. Hughes  
5710 Dellwood Lane  
Beaumont, Texas 77706

```
chkDays(6) = 0  
txtFields(4) = ""  
lblLabels(11) = ""  
Screen.MousePointer = 0
```

```
End Sub
```

```
Sub LoadData()
```

```
With tblAccounts
```

```
txtFields(0) = !([User ID]  
txtFields(1) = !([Account Number]  
txtFields(2) = !([Reference Number]  
txtFields(3) = !([Name]  
If Trim$(![Account Type]) = "Checking" Then Combo1.ListIndex = 0  
If Trim$(![Account Type]) = "Savings" Then Combo1.ListIndex = 1  
If Trim$(![Account Type]) = "CD" Then Combo1.ListIndex = 2  
If Trim$(![Account Type]) = "Loan" Then Combo1.ListIndex = 3  
chkFields(3) = Abs(![Report Balance])  
chkFields(4) = Abs(![Report Transactions])  
chkFields(5) = Abs(![Report NSF] = True)  
chkFields(6) = Abs(![Attach Statement])  
chkFields(1) = Abs(![Extract ASCII])  
chkFields(0) = Abs(![Extract Qwicken])  
chkDays(1) = Abs(!Monday)  
chkDays(2) = Abs(!Tuesday)  
chkDays(3) = Abs(!Wednesday)  
chkDays(4) = Abs(!Thursday)  
chkDays(5) = Abs(!Friday)  
chkDays(6) = Abs(![Monthly Only])
```

12

```
txtFields(4) = !{Monthly Day} & ""
lblLabels(11) = Format$(!{Last Report}, "mm/dd/yyyy")
```

```
End With
```

```
Screen.MousePointer = 0
```

```
On Error Resume Next
```

```
txtFields(1).SetFocus
```

```
On Error GoTo 0
```

```
End Sub
```

```
Private Sub cmdAdd_Click()
```

```
ClearFields
```

```
txtFields(0) = CurrentID
```

```
txtFields(1).SetFocus
```

```
End Sub
```

```
Private Sub cmdDelete_Click()
```

13

End Sub

```
Private Sub cmdPrevious_Click()
    tblAccounts.MovePrevious
    If Err > 0 Or tblAccounts.BOF Then tblAccounts.Seek "<=", CurrentID, 9999999999#
    If tblAccounts![User ID] > CurrentID Then tblAccounts.Seek "<=", CurrentID,
    999999999
    9#
    If tblAccounts![User ID] = CurrentID Then
        ClearFields
        LoadData
    End If
```

End Sub

 Private Sub cmdUpdate\_Click()
 tblAccounts.Seek "=", txtFields(0), txtFields(1)
 If tblAccounts.NoMatch Then tblAccounts.AddNew Else tblAccounts.Edit
 With tblAccounts
 !#[User ID] = txtFields(0)
 !#[Account Number] = txtFields(1)
 !#[Reference Number] = txtFields(2)
 !#[Name] = txtFields(3)
 !#[Account Type] = Combo1
 !#[Report Balance] = (chkFields(3) = 1)
 !#[Report Transactions] = (chkFields(4) = 1)
 !#[Report NSF] = (chkFields(5) = 1)
 !#[Attach Statement] = (chkFields(6) = 1)
 End With

15

```
![Extract ASCII] = (chkFields(1) = 1)
![Extract Qwicken] = (chkFields(0) = 1)
![Monday] = (chkDays(1) = 1)
![Tuesday] = (chkDays(2) = 1)
![Wednesday] = (chkDays(3) = 1)
![Thursday] = (chkDays(4) = 1)
![Friday] = (chkDays(5) = 1)
![Monthly Only] = (chkDays(6) = 1)
If ![Monthly Only] Then
    ![Monthly Day] = Val(txtFields(4))
Else
    ![Monthly Day] = 0
End If
End With
```

frmAccounts - 3

```
tblAccounts.Update
Screen.MousePointer = 0
txtFields(1).SetFocus
```

End Sub

```
Private Sub cmdClose_Click()
Screen.MousePointer = vbDefault
Unload Me
```

16

ClientWidth = 6255  
Icon = (Icon)  
LinkTopic = "Form2"  
LockControls = -1 True  
MaxButton = 0 'False  
MinButton = 0 'False  
ScaleHeight = 4230  
ScaleWidth = 6255  
ShowInTaskbar = 0 'False  
StartUpPosition = 2 'CenterScreen

Begin VB.TextBox txtFields

    DataField = "Monthly Date"  
    DataSource = "datPrimaryRS"  
    Height = 285  
    Index = 4  
    Left = 5400  
    MaxLength = 10  
    TabIndex = 35  
    Top = 3300  
    Width = 315

End

Begin VB.CheckBox chkDays

    Alignment = 1 'Right Justify  
    Caption = "Report on Monday:"  
    DataField = "Monday"  
    DataSource = "datPrimaryRS"  
    Height = 285  
    Index = 1

18

Left = 3660

TabIndex = 34

Top = 1740

Width = 2055

End

Begin VB.CheckBox chkDays

Alignment = 1 'Right Justify

Caption = "Report on Tuesday:"

DataField = "Tuesday"

DataSource = "datPrimaryRS"

Height = 285

Index = 2

Left = 3660

TabIndex = 33

Top = 2055

Width = 2055

End

Begin VB.CheckBox chkDays

Alignment = 1 'Right Justify

Caption = "Report on Wednesday:"

DataField = "Wednesday"

DataSource = "datPrimaryRS"

Height = 285

Index = 3

Left = 3660

TabIndex = 32

Top = 2385

Width = 2055

19

End

frmAccounts - 2

Begin VB.CheckBox chkDays

Alignment = 1 'Right Justify  
Caption = "Report on Thursday."  
DataField = "Thursday"  
DataSource = "datPrimaryRS"  
Height = 285  
Index = 4  
Left = 3660  
TabIndex = 31  
Top = 2700  
Width = 2055

End

Begin VB.CheckBox chkDays

Caption = "Report Monthly only on the      th"  
DataField = "Monthly Only"  
DataSource = "datPrimaryRS"  
Height = 285  
Index = 6  
Left = 3180  
TabIndex = 30  
Top = 3315  
Width = 2775

End

20

Begin VB.CheckBox chkDays

Alignment = 1 'Right Justify  
Caption = "Report on Friday:"  
DataField = "Friday"  
DataSource = "datPrimaryRS"  
Height = 285  
Index = 5  
Left = 3660  
TabIndex = 29  
Top = 3000  
Width = 2055

End

Begin VB.TextBox txtFields

DataField = "Name"  
DataSource = "datPrimaryRS"  
Height = 285  
Index = 3  
Left = 2040  
MaxLength = 50  
TabIndex = 2  
Top = 1080  
Width = 3435

End

Begin VB.TextBox txtFields

DataField = "Reference Number"  
DataSource = "datPrimaryRS"  
Height = 285  
Index = 2

21

Top = 3315

Width = 315

End

Begin VB.ComboBox Combo1

DataField = "Account Type"

DataSource = "datPrimaryRS"

Height = 315

Left = 2040

Style = 2 'Dropdown List'

TabIndex = 3

Top = 1425

Width = 1575

End

Begin VB.PictureBox picButtons

Appearance = 0 'Flat

BorderStyle = 0 'None

ForeColor = &H80000008&

Height = 360

Left = 0

ScaleHeight = 360

ScaleWidth = 6345

TabIndex = 18

Top = 3780

Width = 6345

Begin VB.CommandButton cmdNExt

Caption = "&Next"

Height = 300

Left = 4200

23

Top = 2700

Width = 315

End

Begin VB.CheckBox chkFields

DataField = "Report NSF"

DataSource = "datPrimaryRS"

Height = 285

Index = 5

Left = 2040

TabIndex = 6

Top = 2385

Width = 315

End

Begin VB.CheckBox chkFields

DataField = "Report Transactions"

DataSource = "datPrimaryRS"

Height = 285

Index = 4

Left = 2040

TabIndex = 5

Top = 2055

Width = 315

End

Begin VB.CheckBox chkFields

DataField = "Report Balance"

DataSource = "datPrimaryRS"

Height = 285

Index = 3

Check the View

26

Left = 2040

TabIndex = 4

Top = 1740

Width = 315

End

Begin VB.TextBox txtFields

DataField = "Account Number"

DataSource = "datPrimaryRS"

Height = 285

Index = 1

27

Top = 3000

Width = 1815

End

Begin VB.Label lblLabels

Caption = "Extract Qwicken:"

Height = 255

Index = 7

Left = 120

TabIndex = 23

Top = 3315

Width = 1815

frmAccounts - 6

End

Begin VB.Label lblLabels

Caption = "Attach Statement:"

Height = 255

Index = 6

Left = 120

TabIndex = 17

Top = 2700

Width = 1815

End

Begin VB.Label lblLabels

Caption = "Report NSF:"

30

Height = 255

Index = 5

Left = 120

TabIndex = 16

Top = 2385

Width = 1815

End

Begin VB.Label lblLabels

Caption = "Report Transactions:"

Height = 255

Index = 4

Left = 120

TabIndex = 15

Top = 2055

Width = 1815

End

Begin VB.Label lblLabels

Caption = "Report Balance:"

Height = 255

Index = 3

Left = 120

TabIndex = 14

Top = 1740

Width = 1815

End

Begin VB.Label lblLabels

Caption = "Account Type:"

Height = 255

31

Index = 2  
Left = 120  
TabIndex = 13  
Top = 1425  
Width = 1815

End

Begin VB.Label lblLabels

Caption = "Account Number:"  
Height = 255  
Index = 1  
Left = 120  
TabIndex = 12  
Top = 380  
Width = 1815

End

Begin VB.Label lblLabels

Caption = "Customer ID:"  
Height = 255  
Index = 0  
Left = 120  
TabIndex = 10  
Top = 60  
Width = 1815

End

frmAccounts - 7

End

32

```
tblUsers.MoveFirst  
While Not tblUsers.EOF  
    Combo1.AddItem tblUsers![Sort Name]  
    tblUsers.MoveNext  
Wend  
Combo1.ListIndex = 0  
End Sub
```

frmBroadcast - 1

*E-mail Broadcast*  
VERSION 5.00

```
Object = "{20C62CAE-15DA-101B-B9A8-444553540000}#1.1#0"; "MSMAPI32.OCX"  
Begin VB.Form frmBroadcast  
    BorderStyle = 3 'Fixed Dialog  
    Caption = "e-Mail Broadcast"  
    ClientHeight = 4890  
    ClientLeft = 45  
    ClientTop = 330  
    ClientWidth = 7980  
    Icon = (Icon)  
    LinkTopic = "Form1"  
    LockControls = -1 True  
    MaxButton = 0 'False  
    MinButton = 0 'False  
    ScaleHeight = 4890  
    ScaleWidth = 7980  
    ShowInTaskbar = 0 'False
```

StartUpPosition = 2 'CenterScreen  
Begin MSMAPI.MAPIMessages MAPIMessages1  
    Left       = 2000  
    Top       = 2100  
    \_ExtentX    = 1005  
    \_ExtentY    = 1005  
    \_Version    = 393216  
    AddressEditFieldCount= 1  
    AddressModifiable= 0 'False  
    AddressResolveUI= 0 'False  
    FetchSorted   = 0 'False  
    FetchUnreadOnly = 0 'False

End

Begin MSMAPI.MAPISession MAPISession1  
    Left       = 3240  
    Top       = 2100  
    \_ExtentX    = 1005  
    \_ExtentY    = 1005  
    \_Version    = 393216  
    DownloadMail = 0 'False  
    LogonUI     = -1 True  
    NewSession   = -1 True  
    Password    = "estatement"  
    UserName    = "estatement"

End.

Begin VB.TextBox Text2  
    Height      = 315  
    Left       = 2040

41

```
Index      = 0
Left       = 900
TabIndex   = 5
Top        = 240
Width      = 915
End
End
```

frmGenerate - 1

Option Explicit

Dim FilNum%, CancelFlag%

Sub ReadLoans() 'Read the loan trial balance information

Dim A\$, B\$, C\$, X%, FilNum%, Account\$, FileName\$, Pages%, CurrentPage%,

OldIndex\$

OldIndex\$ = tblAccounts.Index

tblAccounts.Index = "Account Number"

tblBalances.Index = "PrimaryKey"

For X% = 1 To 2

If X% = 1 Then FileName\$ = "Tbal005.prm" Else FileName\$ = "Tbal007.prm"

If Dir(ReportPath & FileName) = "" Then GoTo NextFile

FilNum% = FreeFile

Open ReportPath & FileName For Input As FilNum%

Line Input #FilNum%, A\$

Line Input #FilNum%, A\$

LoanDate = Mid\$(A\$, 47, 2) & "-" & Mid\$(A\$, 49, 2) & "-" & Mid\$(A\$, 51, 2)

LoanDate = Format\$(LoanDate, "mm-dd-yyyy")

Pages% = Val(Mid\$(A\$, 74, 4))

244

CurrentPage% = 0  
Label1 = ""  
Label2 = "Reading " & FileName\$ & "..."  
ProgressBar1.Value = 0  
ProgressBar1.Visible = True  
DoEvents  
While Not EOF(FilNum%)  
Line Input #FilNum%, A\$  
A\$ = RTrim\$(A\$)  
If A\$ = Chr\$(12) Then  
    currentPage% = currentPage% + 1  
    If currentPage% < Pages% Then ProgressBar1.Value = (currentPage% /  
Pages%)  
\* 100  
DoEvents  
GoTo NextLine  
End If  
If Len(A\$) < 80 Then GoTo NextLine  
Account\$ = Trim\$(Mid\$(A\$, 2, 11))  
If Val(Account\$) > 0 Then                 'New Loan  
    tblAccounts.Seek "=", Account\$  
    If tblAccounts.NoMatch = True Then GoTo NextLine  
    Line Input #FilNum%, B\$  
    Line Input #FilNum%, C\$  
    With tblBalances  
        Label1 = "Adding Loan " & Account\$  
        DoEvents  
        .Seek "=", Account\$

45

If .NoMatch Then .AddNew Else .Edit

On Error Resume Next

![Account] = Account\$

![Last Update] = LoanDate

![Balance] = CCur(Trim\$(Mid\$(A\$, 51, 12)))

If IsDate(Mid\$(A\$, 124, 8)) Then ![Date of Last Deposit] = Format\$(Mid\$(A\$. 124. 8))

If IsDate(Mid\$(B\$. 63, 8)) Then ![Maturity Date] = Format\$(Mid\$(B\$, 63, 8), "mm/dd/yyyy")

If IsDate(Mid\$(C\$, 63, 8)) Then ![Next Due Date] = Format\$(Mid\$(C\$, 63, 8), "mm/dd/yyyy")

![Net Payoff] = CCur(Trim\$(Mid\$(C\$, 50, 12)))

![Rate] = CSng(Trim\$(Mid\$(C\$, 76, 7)))

![Payment] = CCur(Trim\$(Mid\$(A\$, 74, 10)))

![YTD Interest] = CCur(Trim\$(Mid\$(C\$, 85, 11)))

![Misc] = Trim\$(Mid\$(B\$, 14, 33))

![Last Deposit] = CCur(Trim\$(Mid\$(B\$, 73, 10))) 'Past Due Amount

If Err = 0 Then .Update Else .CancelUpdate

On Error GoTo 0

End With

fmGenerate - 2

End If

NextLine:

Wend

NextFile:

Close FilNum%

46

e. "Currency") & Chr\$(10)

End If

T\$ = T\$ & " Maturity Date: " & Format\$(tblBalances![Maturity Date], "mm/dd/yyyy") & Chr\$(10)

T\$ = T\$ & " Next Interest Date: " & Format\$(tblBalances![Date of Last Deposit], "mm/dd/yyyy") & Chr\$(10)

If tblBalances![Last Deposit] <> 0 Then T\$ = T\$ & " Accrued Interest: " & Format\$(tblBalances![Last Deposit], "Currency") & Chr\$(10)

T\$ = T\$ & Chr\$(10)

Case "Loan"

If Format\$(LoanDate, "Short Date") <> Format\$(Pdate\$, "Short Date") Then

frmGenerate = 9

T\$ = T\$ & " All Information as of: " & Format\$(LoanDate, "Short Date") & Chr\$(10)

End If

T\$ = T\$ & " Loan Number: " & FNum(tblAccounts![Account Number]) & " - " & tblAccounts!Name & Chr\$(10)

T\$ = T\$ & " Current Balance: " & Format\$(tblBalances![Balance], "Currency") & Chr\$(10)

If tblBalances![Last Deposit] > 0 Then T\$ = T\$ & " Past Due Amount: " & Format\$(tblBalances![Last Deposit], "Currency") & Chr\$(10)

T\$ = T\$ & " Interest Rate: " & Format\$(tblBalances![Rate] / 100, "###.00 %") & Chr\$(10)

T\$ = T\$ & " Payment Amount: " & Format\$(tblBalances![Payment],

11

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag

= False Then GoSub ShowMissing

FileName = "MNTB003.PRN"

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag

= False Then GoSub ShowMissing

FileName = "MNTB004.PRN"

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag

= False Then GoSub ShowMissing

FileName = "MNTB005.PRN"

fmmMain - 6

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag

= False Then GoSub ShowMissing

FileName = "TBAL005.PRN"

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag

= False Then GoSub ShowMissing

FileName = "TBAL007.PRN"

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag

= False Then GoSub ShowMissing

FileName = "TBAL008.PRN"

*List of Report files used for extraction  
to extract routine  
files are available  
list file names*

*this subroutine*

If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
FileName = "PJNL001.PRN"  
If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
FileName = "PJNL002.PRN"  
If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
FileName = "STMT090.PRN"  
If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
FileName = "CFLZ007.PRN"  
If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
FileName = "CF\_\_026.PRN"  
If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
FileName = "CF\_\_027.PRN"  
If Dir\$(ReportPath & FileName) = "" Then FilesMissing = FilesMissing + 1: If  
AutoFlag  
= False Then GoSub ShowMissing  
If FilesMissing = 0 Then

BEST AVAILABLE COPY

92

Left = 240

TabIndex = 3

Top = 0

Width = 975

End

Begin VB.CommandButton cmdClose

Caption = "&Close"

Height = 300

Left = 1350

TabIndex = 1

Top = 0

Width = 975

End

End

Begin VB.Data datPrimaryRS

Align = 2 'Align Bottom

Caption = "

Connect = "Access"

DatabaseName = "C:\ebalance\ebalance.mdb"

DefaultCursorType= 0 'DefaultCursor

DefaultType = 2 'UseODBC

Exclusive = 0 'False

frmUsersGrid - 2

Height = 345

Left = 0

Options = 0

ReadOnly = 0 'False

1540

```
RecordsetType = 1 'Dynaset
RecordSource = ""
Top = 3945
Visible = 0 'False
Width = 8175
End
End
```

Module1 - 1

Option Explicit

Public fMainForm As frmMain

Public CurrentID As Long, LoanDate As Date, LastUpdate As Date

Public ReportPath As String, FilePath As String, LoanPath As String

Public ReportSet As String, ReportDateAdd%

Global wrkJet As Workspace

Global dbs As Database

Global UsersDisabled As Boolean

Global tblAccounts As Table

Global tblUsers As Table

Global tblBalances As Table

Global tblTrans As Table

Global tblNSF As Table

Global tblTranCodes As Table

Global tblHistory As Table

Global AutoFlag%, LastComplete As Date, LastTry As Date, NextTry As Date,

FilesMssin

*Tables for extraneous information*

157

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.  
As rescanning these documents will not correct the image  
problems checked, please do not report these problems to  
the IFW Image Problem Mailbox.**